Optimizing Data Collection for Machine Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Modern deep learning systems require huge data sets to achieve impressive performance, but there is little guidance on how much or what kind of data to collect. Over-collecting data incurs unnecessary present costs, while under-collecting may incur future costs and delay workflows. We propose a new paradigm for modeling the data collection workflow as a formal *optimal data collection problem* that allows designers to specify performance targets, collection costs, a time horizon, and penalties for failing to meet the targets. Additionally, this formulation generalizes to tasks requiring multiple data sources, such as labeled and unlabeled data used in semi-supervised learning. To solve our problem, we develop Learn-Optimize-Collect (LOC), which minimizes expected future collection costs. Finally, we numerically compare our framework to the conventional baseline of estimating data requirements by extrapolating from neural scaling laws. We significantly reduce the risks of failing to meet desired performance targets on several classification, segmentation, and detection tasks, while maintaining low total collection costs.

1 Introduction

When deploying a deep learning model in an industrial application, designers often mandate that the model must meet a pre-determined baseline performance, such as a target metric over a validation data set. For example, an object detector may require a certain minimum mean average precision before being deployed in a safety-critical setting. One of the most effective ways of meeting target performances is by collecting more training data for a given model.

Determining how much data is needed to meet performance targets can impact costs and development delays. Overestimating the data requirement incurs excess costs from collection, cleaning, and annotation. For instance, annotating segmentation masks for a driving data set takes between 15 to 40 seconds per object. For $100{,}000$ images the annotation could require between 170 and 460 days-equivalent of time [1, 2]. On the other hand, collecting too little data may incur future costs and workflow delays from having to collect more later. For example, in medical imaging applications, this means further clinical data acquisition rounds that require expensive clinician time. In the worst case, designers may even realize that a project is infeasible only after collecting insufficient data.

The growing literature on sample complexity in machine learning has identified neural scaling laws that scale model performance with data set sizes according to power laws [3–10]. For instance, Rosenfeld et al. [6] fit power law functions on the performance statistics of small data sets to extrapolate the learning curve with more data. In contrast, Mahmood et al. [2] consider estimating data requirements and show that even small errors in a power law model of the learning curve can translate to massively over- or underestimating how much data is needed. Beyond this, different data sources have different costs and scale differently with performance [11]. For example, although unlabeled data may be easier to collect than labeled data, some semi-supervised learning tasks may need an order of magnitude more unlabeled data to match the performance of a small labeled set. Thus, collecting more data based only on estimation will fail to capture uncertainty and collection costs.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

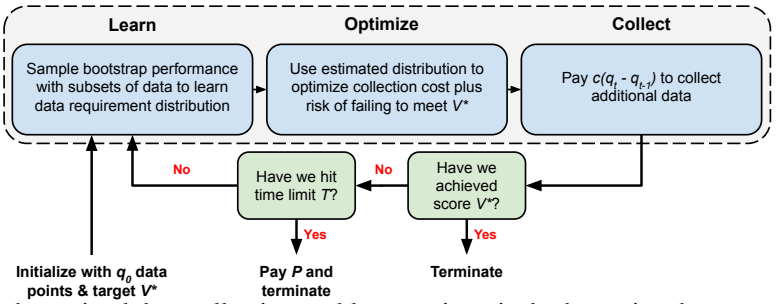

Figure 1: In the optimal data collection problem, we iteratively determine the amount of data that we should have, pay to collect the additional data, and then re-evaluate our model. Our approach, Learn-Optimize-Collect, optimizes for the minimum amount of data q_t^* to collect.

In this paper, we propose a new paradigm for modeling the data collection workflow as an *optimal data collection problem*. Here, a designer must minimize the cost of collecting enough data to obtain a model capable of a desired performance score. They have multiple collection rounds, where after each round, they re-evaluate the model and decide how much more data to order. The data has per-sample costs and moreover, the designer pays a penalty if they fail to meet the target score within a finite horizon. Using this formal framework, we develop an optimization approach for minimizing the expected future collection costs and show that this problem can be optimized in each collection round via gradient descent. Furthermore, our optimization problem immediately generalizes to decisions over multiple data sources (e.g., unlabeled, long-tail, cross-domain, synthetic) that have different costs and impacts on performance. Finally, we demonstrate the value of optimization over naïvely estimating data set requirements (e.g., [2]) for several machine learning tasks and data sets.

Our contributions are as follows. (1) We propose the optimal data collection problem in machine learning, which formalizes data collection workflows. (2) We introduce Learn-Optimize-Collect (LOC), a learning-and-optimizing framework that minimizes future collection costs, can be solved via gradient descent, and has analytic solutions in some settings. (3) We generalize the data collection problem and LOC to a multi-variate setting where different types of data have different costs. To the best of our knowledge, this is the first exploration of data collection with general multiple data sets in machine learning, covering for example, semi-supervised and long-tail learning. (4) We perform experiments over classification, segmentation, and detection tasks to show, on average, approximately a $2\times$ reduction in the chances of failing to meet performance targets, versus estimation baselines.

2 Related work

Neural Scaling Laws. The learning curve and neural scaling law literature argue that model performance (usually defined as validation set loss) scales with the size of the training data set according to a power law function, i.e., $V \propto \theta_0 q^{\theta_1}$ where q is the data set size [5, 6, 8–10, 12–16]. Hestness et al. [5] empirically validate power laws over image classification, language, and audio tasks, while Bahri et al. [9] prove a power law relationship under assumptions on over-parametrization and Lipschitz continuity of the loss, model, and data. Rosenfeld et al. [6] fit power law functions of data set and model size using small training sets and models. Multi-variate scaling laws have also been considered for some specific tasks, for example in transfer learning from synthetic to real data sets [11]. Finally, Mahmood et al. [2] explore data collection by estimating the minimum amount of data needed to meet a given target performance over multiple rounds. Our paper extends these prior studies by developing an optimization problem to minimize the expected total cost of data collected. Specifically, we incorporate the uncertainty in any regression estimate of data requirements and further generalize to multiple data sources with different costs.

Active Learning. Collecting data over multiple rounds is related to active learning [17], where a model selects specifically which data to label from an unlabeled pool when given a fixed labeling budget allocated over multiple rounds of training [18–22]. However, the goal of our work is to systematically determine the optimal collection budget, upon which we may use random sampling or active learning techniques to collect the samples themselves.

Statistical Learning Theory. Accurate theoretical characterizations of the sample complexity of machine learning models may be used to infer data requirements, but the theory is typically only tight

asymptotically. Recent work has explored empirically estimating this theoretical relationship [23, 24]. Bisla et al. [10] study generalization for deep neural networks under assumptions on data set behavior that have some empirical validation. While they highlight use-cases in estimating data requirements from such models, they do not formally explore the consequences of costs of collection.

Optimal Experiment Design. The topic of how to collect data, select samples, and design scientific experiments or controlled trials is well-studied in econometrics [25–27]. For example, Bertsimas et al. [28] optimize the assignment of samples into control and trial groups to minimize inter-group variances. Most recently, Carneiro et al. [29] optimize how many samples and covariates to collect in a statistical experiment by minimizing a treatment effect estimation error or maximizing t-test power. However, our focus on industrial machine learning applications differs from experiment design by having target performance metrics and continual rounds of collection and modeling.

3 Main Problem

In this section, we give a motivating example before introducing the formal data collection problem. We include a table of notation in Appendix A.

Motivating Example. *A startup is developing an object detector for use in autonomous vehicles within the next $T = 5$ years. Their model must achieve a mean Average Precision greater than $V^* = 95\%$ on a pre-determined validation set or else they will lose an expected profit of $P = \$1,000,000$. Collecting training data requires employing drivers to record videos and annotators to label the data, where the marginal cost of obtaining each image is approximately $c = \$1$. In order to manage annual finances, the startup must plan how much data to collect at the beginning of each year.*

Let $z \sim p(z)$ be data drawn from a distribution p. For instance, $z := (x, y)$ may correspond to images x and labels y. Consider a prediction problem for which we train a model with a data set $\mathcal{D}$ of points sampled from $p(z)$. Let $V(\mathcal{D})$ be a score function evaluating the model trained on $\mathcal{D}$.

Optimal Data Collection. We possess an initial data set $\mathcal{D}_{q_0} := \{z_i\}_{i=1}^{q_0}$ of q_0 points; we omit the subscript on $\mathcal{D}$ referring to its size when it is obvious. Our problem is defined by a target score $V^* > V(\mathcal{D}_{q_0})$, a cost-per-sample c of collection, a horizon of T rounds, and a penalty P for failure. At the end of each round $t \in \{1, \dots, T\}$, let q_t be the current amount of data collected. Our goal is to minimize the total cost of collection while building a model that can achieve the target score:

$$\min_{q_1,\dots,q_T} \; c\sum_{t=1}^{T}(q_t - q_{t-1}) + P\mathbb{1}\{V(\mathcal{D}_{q_T}) < V^*\} \qquad \text{s.t. } q_0 \le q_1 \le \cdots \le q_T \tag{1}$$

We collect training data iteratively over multiple rounds (see Figure 1), where in each round, we

1. Decide to grow the data set to $q_t \ge q_{t-1}$ points by sampling $\hat{\mathcal{D}} := \{\hat{z}_i\}_{i=1}^{q_t - q_{t-1}} \sim p(z)$. Pay a cost $c(q_t - q_{t-1})$ and update $\mathcal{D} \leftarrow \mathcal{D} \cup \hat{\mathcal{D}}$.
2. Train the model and evaluate the score. If $V(\mathcal{D}) \ge V^*$, then terminate.
3. If $t = T$, then pay the penalty P and terminate. Otherwise, repeat for the next round.

The model score typically increases monotonically with data set size [5, 6]. This means that the minimum cost strategy for (1) is to collect just enough data such that $V(\mathcal{D}_{q_T}) = V^*$. We can estimate this minimum data requirement by modeling the score function as a stochastic process. Let $V_q := V(\mathcal{D}_q)$ and let $\{V_q\}_{q \in \mathbb{Z}_+}$ be a stochastic process whose indices represent training set sizes in different rounds. Then, collecting data in each round yields a sequence of subsampled data sets $\mathcal{D}_{q_{t-1}} \subset \mathcal{D}_{q_t}$ and their performances $V(\mathcal{D}_{q_t})$. The minimum data requirement is the stopping time

$$D^* := \arg\min_q \{q \mid V_q \ge V^*\}. \tag{2}$$

which is a random variable giving the first time that we pass the target. Note that $q_1^* = \cdots = q_T^* = D^*$ is a minimum cost solution to the optimal data collection problem, incurring a total cost $c(D^* - q_0)$[1].

Estimating D^* using past observations of the learning curve is difficult since we have only T rounds. Further, Mahmood et al. [2] empirically show that small errors in fitting the learning curve can cause massive over- or under-collection. Thus, robust policies must capture the uncertainty of estimation.

[1] We assume that $c(D^* - q_0) < P$, since otherwise the optimal strategy would be to collect no data.

4 Learn-Optimize-Collect (LOC)

Our solution approach, which we refer to as Learn-Optimize-Collect (LOC), minimizes the total collection cost while incorporating the uncertainty of estimating D^*. Although D^* is a discrete random variable, it is realized typically on the order of thousands or greater. To simplify our problem and ensure differentiability, we assume that D^* is continuous and has a well-defined density.

Assumption 1. *The random variable D^* is absolutely continuous and has a cumulative density function (CDF) $F(q)$ and probability density function (PDF) $f(q) := dF(q)/dq$.*

In Section 4.1, we first develop an optimization model when given access to the CDF $f(q)$ and PDF $F(q)$. In Section 4.2, we estimate these distributions and combine them with the optimization model. Finally in Section 4.3, we delineate our optimization approach from prior regression methods.

4.1 Optimization Model

We first propose an optimization problem that at any given round t can simultaneously solve for the optimal amounts of data to collect $q_t, \ldots, q_T$ in all future rounds. Consider the initial setting at $t = 1$. In order to develop intuition, let us first suppose that we know a priori the exact stopping time D^* Then, problem (1) can be re-written as

$$\min_{q_1, \cdots q_T} \; L(q_1, \ldots, q_T; D^*) \qquad \text{s.t. } q_0 \le q_1 \le \cdots \le q_T \tag{3}$$

where the objective function is defined recursively as follows

$$\begin{aligned} L(q_1, \ldots, q_T; D^*) &:= c(q_1 - q_0) + \mathbb{1}\{q_1 < D^*\}\Big(c(q_2 - q_1) + \mathbb{1}\{q_2 < D^*\}\Big(c(q_3 - q_2)\ldots \\ &\qquad \cdots + \mathbb{1}\{q_{T-1} < D^*\}\Big(c(q_T - q_{T-1}) + P\mathbb{1}\{q_T < D^*\}\Big)\cdots\Big)\Big) \\ &= c\sum_{t=1}^{T}(q_t - q_{t-1})\prod_{s=1}^{t-1}\mathbb{1}\{q_s < D^*\} + P\prod_{t=1}^{T}\mathbb{1}\{q_s < D^*\} \\ &= c\sum_{t=1}^{T}(q_t - q_{t-1})\mathbb{1}\{q_{t-1} < D^*\} + P\mathbb{1}\{q_T < D^*\}. \end{aligned}$$

The second line follows from gathering the terms. The third line follows from observing that since $q_1 \le q_2 \le \cdots q_T$ is a constraint, the product of the indicators is equal to the maximum.

In practice, we do not know D^* a priori since it is an unobserved random variable. Instead, suppose we have access to the CDF $F(q)$. Then, we take the expectation over the objective $\mathbb{E}[L(q_1, \ldots, q_T; D^*)]$ to formulate a *stochastic optimization problem* for determining how much data to collect:

$$\min_{q_1, \cdots q_T} \; c\sum_{t=1}^{T}(q_t - q_{t-1})\left(1 - F(q_{t-1})\right) + P\left(1 - F(q_T)\right) \quad \text{s.t. } q_0 \le q_1 \le \cdots \le q_T. \tag{4}$$

Note that the collection variables should be discrete $q_1, \ldots, q_T \in \mathbb{Z}_+$, but similar to the modeling of D^*, we relax the integrality requirement, optimize over continuous variables, and round the final solutions. Furthermore, although problem (4) is constrained, we can re-formulate it with variables $d_t := q_t - q_{t-1}$; this consequently replaces the current constraints with only non-negativity constraints $d_t \ge 0$. Finally due to Assumption 1, problem (6) can be optimized via gradient descent.

4.2 Learning and Optimizing the Data Requirement

Solving problem (4) requires access to the true distribution $F(q)$, which we do not have in reality. In each round, given a current training data set $\mathcal{D}_{q_t}$ of q_t points, we must estimate these distribution functions $F(q)$ and $f(q)$ and then incorporate them into our optimization problem.

Given a current data set $\mathcal{D}_{q_t}$, we may sample an increasing sequence of R subsets $\mathcal{D}_{q_t/R} \subset \mathcal{D}_{2q_t/R} \subset \cdots \subset \mathcal{D}_{q_t}$, fit our model to each subset, and compute the scores to obtain a data set of the learning curve $\mathcal{R} := \{(rq_t/R, V(\mathcal{D}_{rq_t/R}))\}_{r=1}^{R}$. In order to model the distribution of D^*, we can take B bootstrap resamples of $\mathcal{R}$ to fit a series of regression functions and obtain corresponding estimates

$\{\hat{D}_b\}_{b=1}^B$. Given a set of estimates of the data requirement, we then estimate the probability density function via Kernel Density Estimation. Finally to fit the CDF, we numerically integrate the PDF.

In our complete framework, LOC, we first estimate $F(q)$ and $f(q)$. We then use these models to solve problem (4). Note that in the t-th round of collection, we fix the prior decision variables $q_1, \dots q_{t-1}$ constant. Finally, we collect data as determined by the optimal solution q_t^* to problem (4). Full details of the learning and optimization steps, including the complete Algorithm, are in Appendix B.

4.3 Comparison to Mahmood et al. [2]

Our prediction model extends the previous approach of Mahmood et al. [2], who consider only point estimation of D^*. They (i) build the set $\mathcal{R}$, (ii) fit a parametric function $\hat{v}(q;\boldsymbol{\theta})$ to $\mathcal{R}$ via least-squares minimization, and (iii) solve for $\hat{D} = \arg\min_q\{q \mid \hat{v}(q;\boldsymbol{\theta}) \geq V^*\}$. They use several parametric functions from the neural scaling law literature, including the power law function, $\hat{v}(q;\boldsymbol{\theta}) := \theta_0 q^{\theta_1} + \theta_2$ [8, 2], and use an ad hoc correction factor obtained by trial and error on past tasks to help decrease the failure rate. Instead, we take bootstrap samples of $\mathcal{R}$ to fit multiple regression functions, estimate a distribution for $\hat{D}$, and incorporate them into our novel optimization model. Finally, we show in the next two sections that our optimization problem has analytic solutions and extends to multiple sources.

5 Analytic Solutions for the $T = 1$ Setting

In this section, we explore analytic solutions for problem (4). The unobservable D^* and sequential decision-making nature suggest this problem can be formulated as a Partially Observable Markov Decision Process (POMDP) with an infinite state and action space (see Appendix C.1), but such problems rarely permit exact solution methods [30]. Nonetheless, we can derive exact solutions for the simple case of a single $T = 1$ round, re-stated below

$$\min_{q_1} \; c(q_1 - q_0) + P(1 - F(q_1)) \qquad \text{s. t. } q_0 \leq q_1 \tag{5}$$

Theorem 1. *Assume $F(q)$ is strictly increasing and continuous. For any ϵ such that $F(q_0) < 1-\epsilon$, let $P := c/f(F^{-1}(1-\epsilon))$. The optimal solution to the corresponding problem* (5) *is $q_1^* = F^{-1}(1-\epsilon)$. Furthermore, this solution satisfies $F(q_1^*) = 1 - \epsilon$.*

When the penalty P is specified via a failure risk ϵ, the optimal solution to problem (5) is equal to a quantile of the distribution of D^*. We defer the proof and some auxiliary results to Appendix C.2.

Theorem 1 further provides guidelines on choosing values for the cost and penalty parameters. While c is the dollar-value cost per-sample, which includes acquisition, cleaning, and annotation, P can reflect their inherent regret or opportunity cost of failing to meet their target score. A designer can accept a risk ϵ of failing to collect enough data $\Pr\{q^* < D^*\} = \epsilon$. From Theorem 1, their optimal strategy should be to collect $F^{-1}(1-\epsilon)$ points, which is also the optimal solution to problem (5).

6 The Multi-variate LOC: Collecting Data from Multiple Sources

So far, we have assumed that a designer only chooses how much data to collect and must pay a fixed per-sample collection cost. We now explore the multi-variate extension of the data collection problem where there are different types of data with different costs. For example, consider long-tail learning where samples for some rare classes are harder to obtain and thus, more expensive [31], semi-supervised learning where labeling data may cost more than collecting unlabeled data [32], or domain adaptation where a source data set is easier to obtain than a target set [33]. In this section, we highlight our main formulation and defer the complete multi-variate LOC to Appendix D.

Consider $K \in \mathbb{N}$ data sources (e.g., $K = 2$ with labeled and unlabeled) and for each $k \in \{1, \dots, K\}$, let $z^k \sim p_k(z^k)$ be data drawn from their distribution. We train a model with a data set $\mathcal{D} := \cup_{k=1}^K \mathcal{D}^k$ where each $\mathcal{D}^k$ contains points of the k-th source. The performance or score function of our model is $V(\mathcal{D}^1, \dots, \mathcal{D}^K)$. For each k, we initialize with q_0^k points. Let $\mathbf{q}_0 = (q_0^1, \dots, q_0^K)^\mathsf{T}$ denote the vector of data set sizes and let $\mathbf{c} = (c^1, \dots, c^K)^\mathsf{T}$ denote costs (i.e., c^k is the cost of collecting data from

$p_k(z^k)$). Given a target V^*, penalty P, and T rounds, we want to minimize the total cost of collection

$$\min_{\mathbf{q}_1,\ldots,\mathbf{q}_T} \mathbf{c}^\mathsf{T} \sum_{t=1}^{T} (\mathbf{q}_t - \mathbf{q}_{t-1}) + P\mathbb{1}\{V(\mathcal{D}_{q_T^1}, \ldots, \mathcal{D}_{q_T^K}) < V^*\} \qquad \text{s.t. } \mathbf{q}_0 \leq \mathbf{q}_1 \leq \mathbf{q}_2 \leq \cdots \leq \mathbf{q}_T$$

We can follow the same steps shown in Section 4 to solve this problem. First, the learning curve is now a stochastic process $\{V_\mathbf{q}\}_{\mathbf{q}\in\mathbb{Z}_+^K}$ indexed in K dimensions. Further, the multi-variate analogue of the minimum data requirement in (2) is now a vector that states the minimum cost amount of data needed to meet the target score:

$$\mathbf{D}^* := \arg\min_{\mathbf{q}} \left\{ \mathbf{c}^\mathsf{T}\mathbf{q} \mid V_\mathbf{q} \geq V^* \right\}$$

We randomly pick a unique solution to break ties. From Assumption 1, $\mathbf{D}^*$ is a random vector with a PDF $f(\mathbf{q})$ and a CDF $F(\mathbf{q}) := \int_\mathbf{0}^\mathbf{q} f(\hat{\mathbf{q}}) d\hat{\mathbf{q}}$. Finally, the multi-variate analogue of the stochastic problem (4) is

$$\min_{\mathbf{q}_1,\cdots,\mathbf{q}_T} \mathbf{c}^\mathsf{T} \sum_{t=1}^{T} (\mathbf{q}_t - \mathbf{q}_{t-1}) \left(1 - F(\mathbf{q}_{t-1})\right) + P\left(1 - F(\mathbf{q}_T)\right) \text{ s.t. } \mathbf{q}_0 \leq \mathbf{q}_1 \leq \cdots \leq \mathbf{q}_T \qquad (6)$$

The Multi-variate LOC requires multi-variate PDFs, which we can fit in the same way as discussed in Section 4.2. However, we now need multi-variate regression functions that can accommodate different types of data. In Appendix D, we propose an additive family of power law regression functions that can handle an arbitrary number of K sources. In our experiments, we also generalize the estimation approach of Mahmood et al. [2] to the multi-source setting for comparison.

7 Empirical Results

We explore the data collection problem over two sets of experiments covering single-variate $K = 1$ (Section 4) and multi-variate $K = 2$ (Section 6) problems. We consider image classification, segmentation, and object detection tasks. For every data set and task, LOC significantly reduces the number of instances where we fail to meet a data requirement V^*, while incurring a competitive cost with respect to the conventional baseline of naïvely estimating the data requirement [2].

In this section, we summarize the main results. We detail our data collection and experiment setup in Appendix E. We expand our full results in Appendix F.

7.1 Data and Methods

When $K = 1$, the designer decides how much data to sample without controlling the type of data. We explore classification on CIFAR-10 [34], CIFAR-100 [34], and ImageNet [35], where we train ResNets [36] to meet a target validation accuracy. We explore semantic segmentation using Deeplabv3 [37] on BDD100K [38], which is a large-scale driving data set, as well as Bird's-Eye-View (BEV) segmentation on nuScenes [39] using the 'Lift Splat' architecture [40]; for both tasks, we desire a target mean intersection-over-union (IoU). We explore 2-D object detection on PASCAL VOC [41, 42] using SSD300 [43], where we evaluate mean average precision (mAP).

When $K = 2$, the designer collects two types of data with different costs. We first divide CIFAR-100 into two subsets containing data from the first and last 50 classes, respectively. Here, we assume that the first 50 classes are more expensive to collect than the last; this mimics a real-world scenario where collecting data for some classes (e.g., long-tail) is more expensive than others. We then explore semi-supervised learning on BDD100K where the labeled subset of this data is more expensive than the unlabeled data; the cost difference between these two types is equal to the cost of data annotation.

We use a simulation model of the deep learning workflow following the procedure of Mahmood et al. [2], to approximate the true problem while simplifying the experiments (see Appendix E for full details). To avoid repeatedly sampling data, re-training a model, and evaluating the score, each simulation uses a piecewise-linear approximation of a 'ground truth' learning curve that returns model performance as a function of data set size. In our problems, we initialize with $q_0 = 10\%$ of the full data set (we use 20% for VOC). Then in each round, we solve for the amount of data to collect and then call the piecewise-linear learning curve to obtain the current score.

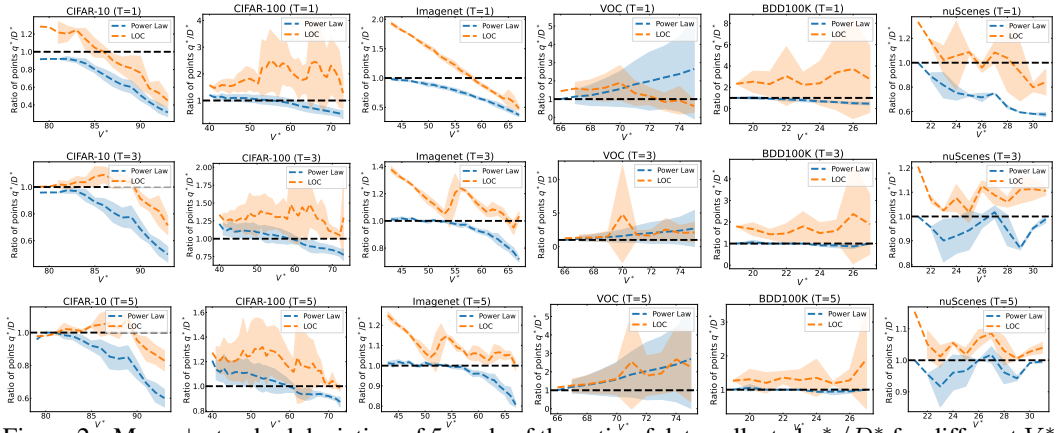

Figure 2: Mean $\pm$ standard deviation of 5 seeds of the ratio of data collected q_T^*/D^* for different V^*. The rows correspond to $T = 1, 3, 5$ and the columns to different data sets. The black line corresponds to collecting exactly the minimum data requirement. LOC consistently remains slightly above the black line, meaning we rarely fail to meet the target.

	Data set	T	Power Law Regression		LOC	
			Failure rate	Cost ratio	Failure rate	Cost ratio
Class.	CIFAR-10	1	100%	–	**60%**	0.19
		3	95%	0.00	**32%**	0.05
		5	86%	0.00	**29%**	0.03
	CIFAR-100	1	56%	0.12	**4%**	0.99
		3	48%	0.10	**3%**	0.31
		5	48%	0.10	**2%**	0.19
	Imagenet	1	99%	0.00	**37%**	0.49
		3	75%	0.01	**5%**	0.16
		5	56%	0.01	**2%**	0.10
Seg.	BDD100K	1	77%	0.03	**12%**	2.03
		3	31%	0.00	**0%**	0.72
		5	23%	0.01	**0%**	0.35
	nuScenes	1	95%	0.00	**52%**	0.16
		3	71%	0.01	**0%**	0.09
		5	62%	0.00	**0%**	0.04
Det.	VOC	1	36%	1.24	**25%**	0.56
		3	8%	0.88	**0%**	1.10
		5	6%	0.86	**0%**	0.84

Table 1: Average cost ratio $\mathbf{c}^\mathsf{T}(\mathbf{q}_T^* - \mathbf{q}_0)/\mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0) - 1$ and failure rate measured over a range of V^* for each T and data set. We fix $c = 1$ and $P = 10^7$ ($P = 10^6$ for VOC and $P = 10^8$ for ImageNet). The best performing failure rate for each setting is bolded. The cost ratio is measured only for instances that achieve V^*. LOC consistently reduces the average failure rate, often down to 0%, while keeping the average cost ratio almost always below 1 (i.e., spending at most $2\times$ the optimal amount).

We compare LOC against the conventional estimation approach of Mahmood et al. [2], who fit a regression model to the learning curve statistics, extrapolate the learning curve for larger data sets, and then solve for the minimum data requirement under this extrapolation. There are many different regression models that can be used to fit learning curves [12, 14, 5, 8], but power laws are the most commonly studied approach in the neural scaling law literature. Consequently, we use power law regression to model the learning curve for the baseline and LOC.

7.2 Main Results

We consider $T = 1, 3, 5$ rounds and $V^* \in [V(\mathcal{D}_{q_0}) + 1, V(\mathcal{D})]$ targets, where $\mathcal{D}$ is the entire data set. We evaluate all methods on (i) the failure rate, which is how often the method fails to achieve the given V^* within T rounds, and (ii) the cost ratio, which is equal to $\mathbf{c}^\mathsf{T}(\mathbf{q}_T^* - \mathbf{q}_0)/\mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0) - 1$. For $K = 1$, we also measure the ratio of points collected q_T^*/D^*. Although there is a natural trade-off between low cost ratio (under-collecting) and failure rate (over-collecting), we emphasize that our goal is to have low cost but with zero chance of failure.

The Value of Optimization over Estimation when $K = 1$. Figure 2 compares LOC versus the corresponding power law regression baseline when $c = 1$ and $P = 10^7$ ($P = 10^6$ for VOC and $P = 10^8$ for ImageNet). If a curve is below the black line, then it failed to collect enough data to meet the target. LOC consistently remains above this black line for most settings. In contrast, even with up to $T = 5$ rounds, collecting data based only on regression estimates leads to failure.

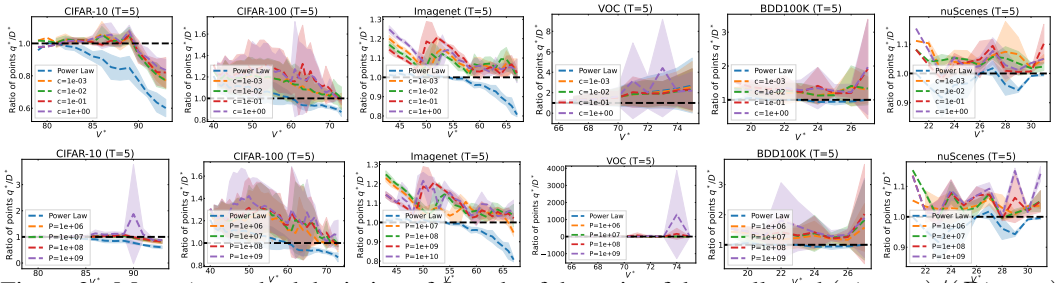

Figure 3: Mean $\pm$ standard deviation of 5 seeds of the ratio of data collected $(q_T^* - q_0)/(D^* - q_0)$ for different V^* and fixed $T = 5$. *Top:* We sweep the cost parameter from 0.001 to 1 and fix $P = 10^7$. *Bottom:* We sweep the penalty parameter from 10^6 to 10^9 and fix $c = 1$. The dashed black line corresponds to collecting exactly the minimum data requirement. See Appendix F for all T.

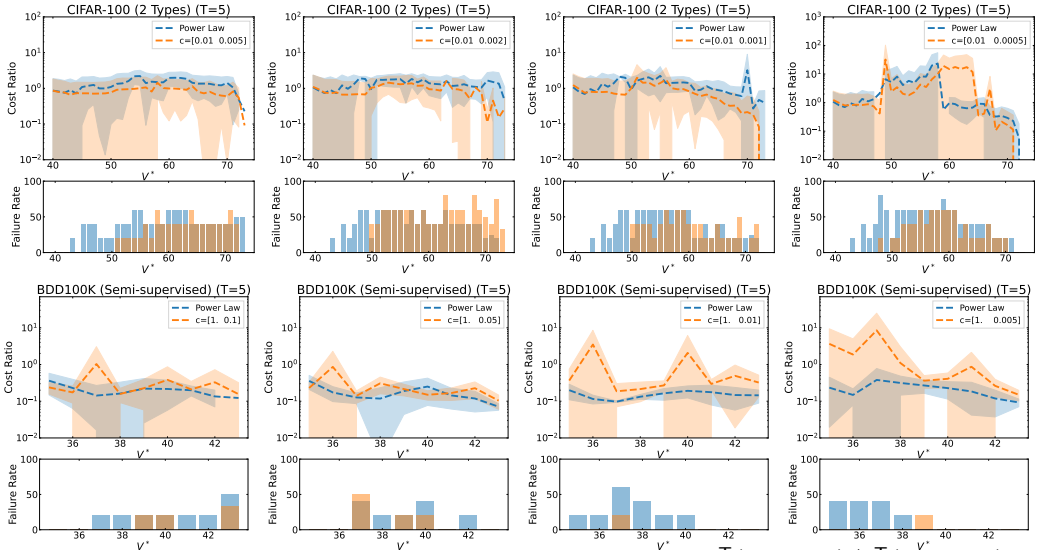

Figure 4: Mean $\pm$ standard deviation over 5 seeds of the cost ratio $\mathbf{c}^\mathsf{T}(\mathbf{q}_T^* - \mathbf{q}_0)/\mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0) - 1$ and failure rate for different V^*, after removing 99-th percentile outliers. The columns correspond to scenarios where the first set c^1 costs increasingly more than the second c^2. See Appendix F for all T.

Table 1 aggregates the failure rates and cost ratios for each setting. To summarize, LOC fails at less than 10% of instances for $12/18$ settings, whereas regression fails over 30% for $15/18$ settings. In particular, regression nearly always under-collects data when given a single $T = 1$ round. Here, LOC reduces the risk of under-collecting by 40% to 90% over the baseline. While this leads to a marginal increase in costs, our cost ratios are consistently less than 0.5 for $12/18$ settings, meaning that we spend at most 50% more than the true minimum cost.

We remark that previously, Mahmood et al. [2] observed that incorrect regression estimates necessitated real machine learning workflows to collect data over multiple rounds. Instead, with LOC, we can make significantly improved data collection decisions even with a single round.

Robustness to Cost and Penalty Parameters (see Appendix F.1 for details). Figure 3 evaluates the ratio of points collected for $T = 5$ when the cost and the penalty of the optimization problem are varied. Our algorithm is robust to variations in these parameters, as LOC retain the same shape and scale for almost every parameter setting and data set. Further, LOC consistently remain above the horizontal 1 line, showing that even after varying c and P, we do not fail as frequently as the baseline. Finally, validating Theorem 1, the penalty parameter P provides natural control over the amount of data collected. As we increase P, the ratio of data collected increases consistently.

The Value of Optimization over Estimation when $K = 2$ (Appendix F.2). Figure 4 compares LOC versus regression at $T = 5$ with different costs, showing that we maintain a similar cost ratio to the regression alternative, but with lower failure rates. Table 2 aggregates failure rates and cost ratios for all settings, showing LOC consistently achieves lower failure rates for nearly all settings of T. When $T = 5$, LOC also achieves lower cost ratios versus regression on CIFAR-100, meaning that

Data set	T	Cost	Power Law Regression		LOC	
			Failure rate	Cost ratio	Failure rate	Cost ratio
CIFAR-100 (2 Types)	1	$(0.01, 0.0005)$	62%	0.89	**40**%	41.80
		$(0.01, 0.001)$	58%	1.19	**46**%	9.85
		$(0.01, 0.002)$	56%	1.55	**54**%	6.98
		$(0.01, 0.005)$	54%	1.65	**33**%	4.43
	3	$(0.01, 0.0005)$	43%	3.47	**30**%	4.88
		$(0.01, 0.001)$	45%	1.22	**43**%	1.31
		$(0.01, 0.002)$	45%	1.47	**44**%	1.21
		$(0.01, 0.005)$	38%	1.31	**36**%	1.17
	5	$(0.01, 0.0005)$	38%	3.31	**24**%	5.19
		$(0.01, 0.001)$	35%	1.22	**24**%	0.79
		$(0.01, 0.002)$	**37**%	1.33	38%	0.90
		$(0.01, 0.005)$	36%	1.30	**24**%	0.82
BDD100K (Semi-supervised)	1	$(1, 0.005)$	86%	0.11	**44**%	7.02
		$(1, 0.01)$	79%	0.15	**30**%	13.47
		$(1, 0.05)$	72%	0.19	**49**%	1.02
		$(1, 0.1)$	70%	0.19	**65**%	0.40
	3	$(1, 0.005)$	23%	0.18	**7**%	1.20
		$(1, 0.01)$	21%	0.15	**7**%	2.57
		$(1, 0.05)$	26%	0.18	**23**%	0.50
		$(1, 0.1)$	26%	0.21	**30**%	0.15
	5	$(1, 0.005)$	16%	0.22	**2**%	1.91
		$(1, 0.01)$	21%	0.15	**2**%	0.86
		$(1, 0.05)$	16%	0.17	**9**%	0.27
		$(1, 0.1)$	16%	0.20	**7**%	0.32

Table 2: Average cost ratio $\mathbf{c}^\mathsf{T}(\mathbf{q}^*_T - \mathbf{q}_0)/\mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0) - 1$ and failure rate over different V^* for each T and $\mathbf{c}$, after removing 99-th percentile outliers. We fix $P = 10^{13}$ for CIFAR-100 and $P = 10^8$ for BDD100K. The best performing failure rate for each setting is bolded. The cost ratio is measured over instances that achieve V^*. LOC consistently reduces the average failure rate, and for $T > 1$, preserves the cost ratio. Further, LOC is more robust to uneven costs than regression.

with multiple rounds of collection, we can ensure meeting performance requirements while paying nearly the optimal amount of data. However, solving the optimization problem is generally more difficult as K increases, and we sometimes over-collect data by large margins; consequently, we report these results after removing the 99-th percentile outliers with respect to total cost for both methods. Nonetheless, this challenge remains when $T = 1$, particularly for CIFAR-100.

8 Discussion

We develop a rigorous framework for optimizing data collection workflows in machine learning applications, by introducing an optimal data collection problem that captures the uncertainty in estimating data requirements. We generalize this problem to more realistic settings where multiple data sources incur varying costs of collection. We validate our solution algorithm, LOC, on six data sets covering classification, segmentation, and detection tasks to show that we consistently meet pre-determined performance metrics regardless of costs and time horizons.

Our approach relies on estimating the CDF and PDF of the minimum data requirement, which is a challenging problem, especially with multiple data sources. Nonetheless, LOC can be deployed on top of future advances in estimating neural scaling laws. Further, we allow practitioners to input problem-specific costs and penalties, but these quantities may not always be readily available. We provide some theoretical insight into parameter selection and show that LOC is robust to these parameters. Finally, our empirical analysis focuses on computer vision, but we expect our approach to be viable in other domains governed by scaling laws.

Improving data collection practices yields potentially positive and negative societal impacts. LOC reduces the collection of extraneous data, which can, in turn, reduce the environmental costs of training models. On the other hand, equitable data collection should also be considered in real-world data collection practices that involve humans. We envision a potential future work to incorporate privacy and fairness constraints to prevent over- or under-sampling of protected groups. Finally, our method is guided by a score function on a held-out validation set. Biases in this set may be exacerbated when optimizing data collection to meet target performance.

There is a folklore observation that over 80% of industry machine learning projects fail to reach production, often due to insufficient, noisy, or inappropriate data [44, 45]. Our experiments verify this by showing that naïvely estimating data requirements will often yield failures to meet target performances. We believe that robust data collection policies obtained via LOC can reduce failures while further guiding practitioners on how to manage both costs and time.

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] Rafid Mahmood, James Lucas, David Acuna, Daiqing Li, Jonah Philion, Jose M. Alvarez, Zhiding Yu, Sanja Fidler, and Marc T. Law. How much more data do we need? estimating requirements for downstream tasks. In *2022 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2022.

[3] Lewis J Frey and Douglas H Fisher. Modeling decision tree performance with the power law. In *Seventh International Workshop on Artificial Intelligence and Statistics*. PMLR, 1999.

[4] Baohua Gu, Feifang Hu, and Huan Liu. Modelling classification performance for large data sets. In *International Conference on Web-Age Information Management*, pages 317–328. Springer, 2001.

[5] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

[6] Jonathan S Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations*, 2020.

[7] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[8] Derek Hoiem, Tanmay Gupta, Zhizhong Li, and Michal Shlapentokh-Rothman. Learning curves for analysis of deep networks. In *International Conference on Machine Learning*, pages 4287–4296. PMLR, 2021.

[9] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *arXiv preprint arXiv:2102.06701*, 2021.

[10] Devansh Bisla, Apoorva Nandini Saridena, and Anna Choromanska. A theoretical-empirical approach to estimating sample complexity of dnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3270–3280, 2021.

[11] Hiroaki Mikami, Kenji Fukumizu, Shogo Murai, Shuji Suzuki, Yuta Kikuchi, Taiji Suzuki, Shin-ichi Maeda, and Kohei Hayashi. A scaling law for synthetic-to-real transfer: How much is your pre-training effective? *arXiv preprint arXiv:2108.11018*, 2021.

[12] S Jones, S Carley, and M Harrison. An introduction to power and sample size estimation. *Emergency Medicine Journal: EMJ*, 20(5):453, 2003.

[13] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 843–852, 2017.

[14] Rosa L Figueroa, Qing Zeng-Treitler, Sasikiran Kandula, and Long H Ngo. Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making*, 12(1):1–10, 2012.

[15] Tom Viering and Marco Loog. The shape of learning curves: a review. *arXiv preprint arXiv:2103.10948*, 2021.

[16] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[17] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[18] Burr Settles. Active learning literature survey. 2009.

[19] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

[20] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 93–102, 2019.

[21] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.

[22] Rafid Mahmood, Sanja Fidler, and Marc T. Law. Low-budget active learning via wasserstein distance: An integer programming approach. In *International Conference on Learning Representations*, 2022.

[23] Yiding Jiang, Pierre Foret, Scott Yak, Daniel M Roy, Hossein Mobahi, Gintare Karolina Dziugaite, Samy Bengio, Suriya Gunasekar, Isabelle Guyon, and Behnam Neyshabur. Neurips 2020 competition: Predicting generalization in deep learning. *arXiv preprint arXiv:2012.07976*, 2020.

[24] Yiding Jiang, Parth Natekar, Manik Sharma, Sumukh K Aithal, Dhruva Kashyap, Natarajan Subramanyam, Carlos Lassance, Daniel M Roy, Gintare Karolina Dziugaite, Suriya Gunasekar, et al. Methods and analysis of the first competition in predicting generalization of deep learning. In *NeurIPS 2020 Competition and Demonstration Track*, pages 170–190. PMLR, 2021.

[25] Kirstine Smith. On the standard deviations of adjusted and interpolated values of an observed polynomial function and its constants and the guidance they give towards a proper choice of the distribution of observations. *Biometrika*, 12(1/2):1–85, 1918.

[26] David Cohn. Neural network exploration using optimal experiment design. *Advances in neural information processing systems*, 6, 1993.

[27] Ashley F Emery and Aleksey V Nenarokomov. Optimal experiment design. *Measurement Science and Technology*, 9(6):864, 1998.

[28] Dimitris Bertsimas, Mac Johnson, and Nathan Kallus. The power of optimization over randomization in designing experiments involving small samples. *Operations Research*, 63(4):868–876, 2015.

[29] Pedro Carneiro, Sokbae Lee, and Daniel Wilhelm. Optimal data collection for randomized control trials. *The Econometrics Journal*, 23(1):1–31, 2020.

[30] Hao Zhang. Dynamic learning and decision making via basis weight vectors. *Operations Research*, 2022.

[31] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications*, 73:220–239, 2017.

[32] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109 (2):373–440, 2020.

[33] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[34] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[37] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[38] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020.

[39] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

[40] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Proceedings of the European Conference on Computer Vision*, 2020.

[41] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, .

[42] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, .

[43] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[44] Rob van der Meulen and Thomas McCall. Gartner says nearly half of cios are planning to deploy artificial intelligence, Feb 2018. URL `https://www.gartner.com/en/newsroom/press-releases/2018-02-13-gartner-says-nearly-half-of-cios-are-planning-to-deploy-artificial-intelligence`.

[45] Why do 87% of data science projects never make it into production?, Jul 2019. URL `https://venturebeat.com/2019/07/19/why-do-87-of-data-science-projects-never-make-it-into-production/`.

[46] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[47] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[48] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[49] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

[50] David Easley and Nicholas M Kiefer. Controlling a stochastic process with unknown parameters. *Econometrica: Journal of the Econometric Society*, pages 1045–1064, 1988.

[51] Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

[52] Eric Zhao, Anqi Liu, Animashree Anandkumar, and Yisong Yue. Active learning under label shift. In *International Conference on Artificial Intelligence and Statistics*, pages 3412–3420. PMLR, 2021.

[53] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.

[54] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2020.

[55] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[56] Ismail Elezi, Zhiding Yu, Anima Anandkumar, Laura Leal-Taixe, and Jose M Alvarez. Not all labels are equal: Rationalizing the labeling costs for training object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See the Conclusion.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See the Conclusion.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Assumption 1 and the theorem statements.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix C.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code is proprietary. The data is publicly available. See Algorithm 1 and Appendix F for implementation details.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix E and F for details.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All experiments were over five seeds. See the for standard deviation ranges.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix E.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite all data sets and model architectures in Appendix E.
   (b) Did you mention the license of the assets? [N/A] All data sets and models are publicly available.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]